TECHNIQUES FOR THE INTERACTIVE EXPLORATION OF HIGH-DETAIL 3D BUILDING RECONSTRUCTIONS USING THE EXAMPLE OF ROMAN COLOGNE

S. Maass a, *, M. Trapp a, J.E. Kyprianidis a, J. Döllner a, M. Eichhorn b, R. Pokorski b, J. Bäuerlein c, H. v. Hesberg c

[a] Dep. of Computer Graphics, University of Potsdam, HPI, Germany - (maass, trapp, kyprianidis, doellner)@hpi.uni-potsdam.de
[b] KISD, University of Applied Sciences Cologne, Germany - (michael.eichhorn , rafael.pokorski)@kisd.de
[c] Archaeological Institute, University of Cologne, Germany - j.baeuerlein@uni-koeln.de, hesberg@rom.dainst.org

KEY WORDS: High-detail 3D Models, Virtual Reality, Real-Time 3D Visualization, Roman Cologne

ABSTRACT:

This paper presents the results achieved by an interdisciplinary team of archaeologists, designers, and computer graphics engineers with the aim to virtually reconstruct an interactive high-detail 3D city model of Roman Cologne. We describe a content creation pipeline established to enable a flexible exchange and enhancement of building models, the applied optimization techniques necessary for real-time rendering, and the design of an application framework that enables the coupling of 3D visualizations with additional information in corresponding Adobe® Flash® widgets. Furthermore, we expose challenges arising by incorporating state-of-the-art visualization techniques, such as cut-away views, non-photorealistic rendering (NPR), and automated label placement. These techniques are used to enhance the interactive 3D environments, to enable for the presentation of interior structures, the precise communication what is hypothetic and what proven knowledge, and the integration of meta-information.

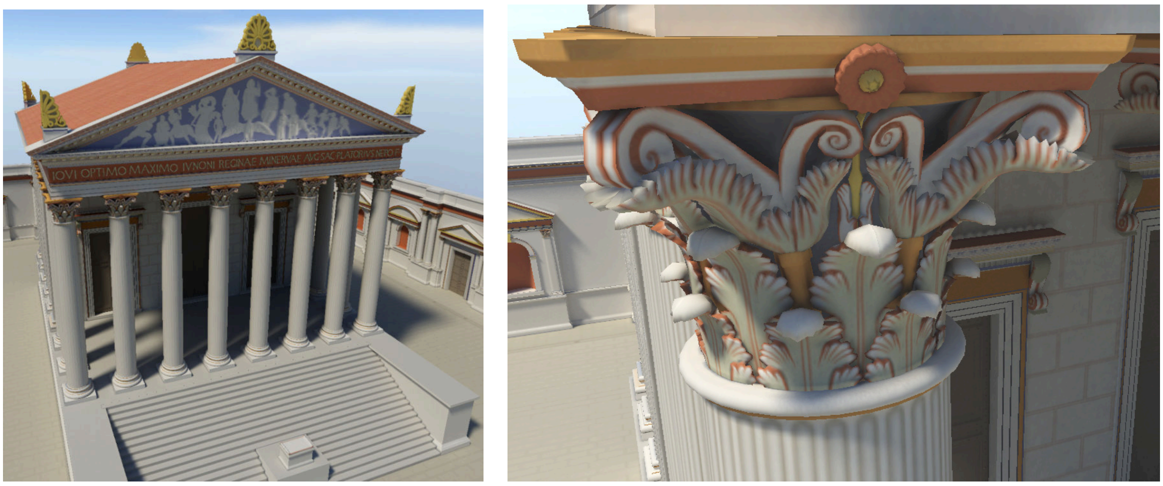

Figure 1: Screenshots from the interactive 3D visualization of Roman Cologne: temple and a detailed view of a capital.

1. INTRODUCTION

Virtual 3D reconstructions of archaeological excavation sites, artefacts, and architecture play an important role preserving our cultural heritage for future generations. Used in combination with interactive visualization technology, they become a powerful tool to support scientific discussions among experts and to present important archaeological facts to broad audiences using museum and edutainment applications. However, the creation and interactive exploration of high-detailed 3D reconstructions is still a challenge for the community (Santos et al. 2006, Kuchar et al. 2007) and most projects finally result in still images, pre-rendered animation videos, or QuickTime® panoramas as a compromise between visual quality and interactivity (e.g., Debevec, P. E., 2005, Almagro et al. 2006).

As an interdisciplinary team, consisting of archaeologists, designers, and computer graphic engineers, the project „*Visualization of Roman Cologne*'' started, with the vision to overcome these limitations. Our aim was to construct high-detail virtual 3D models and a visualization framework that enables their interactive exploration and presentation (Figure 1). This paper presents the results of the combined expertise of all teams. It can be read as a guideline for similar future projects, e.g., to setup a collaborative content creation process, select appropriate data exchange formats, or to apply the presented visualization and optimization techniques to other domains of virtual archaeology.

The paper is structured as follows: Section 2 describes the complete creation process from archaeological reconstructions to the 3D visualizations. Section 3 discusses the optimizations of geometric complex reconstructed building models to increase their applicability for an interactive visualization. Section 4 presents the flexibility of our framework that is completely configurable via XML descriptions and can be coupled with Adobe® Flash® widgets to present secondary information using multimedia content. Section 5 presents enhanced visualization techniques that can be used to ease the communication of specific archaeological aspects to the user. Finally, Section 6 summarizes the results and gives an outlook to future work.

2. CONTENT CREATION PIPELINE

To start the collaborative work, we first setup a content creation pipeline, define the data exchange formats, possible use-cases, and the roles for each team. This process should guarantee that all experts within the teams can operate in their domains without restrictions. Additionally, the pipeline is designed to facilitate three important aspects:

1. **Content preservation**: The representation and encoding formats of the accumulated primary (3D models) and secondary data (multimedia content) has to ensure its accessibility for future usage.
2. **Extensibility**: The established pipeline should support the integration of upcoming ideas and new technology in all its stages during the project.
3. **Re-usability**: The developed framework should be robust, flexible, and easily adaptable to enable its application in other interdisciplinary visualization projects.

Figure 2 illustrates the workflow that is described in detail as follows.

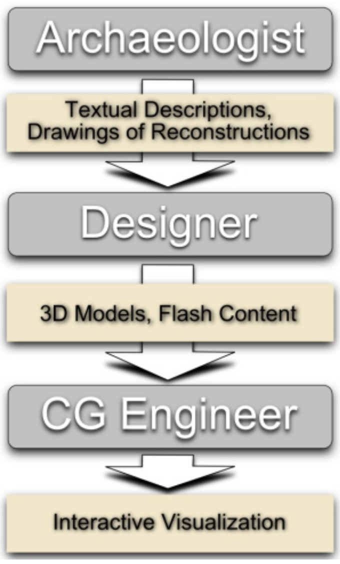

Figure 2: Illustration of the simplified content creation process with the involved teams and the data flow.

2.1 Scientific Reconstruction

To create a complete 3D model representing the ancient town of Cologne, a number of single roman structures and building elements have to be reconstructed first. The reconstruction of these elements, their combination, and arrangement was done by the archaeology experts. For this task, known facts from previous research, results of actual publications (e.g., Irmler 2004), as well as recent finds of the local department of antiquities of the Romano-Germanic Museum Cologne were considered. Only well-studied buildings, with a scientifically evaluated shape or floor plan, were reconstructed in high detail. Thereby, analogies to reconstructions, done before, were used to derive new reliable 3D models. For all other elements only simple shapes were selected for the reconstructions to communicate the missing evidence.

Additionally, a digital terrain model (DTM) (Weibel and Heller 1991) of the ancient Cologne with building sites and streets was reconstructed to embed the 3D buildings later on. It was derived from a DTM of the present Cologne, whose geo-reference was adapted to the location of finds as well as to known morphological changes in the past. Furthermore, the ancient settlement area was bounded by a plateau and a city wall; structures that can be partially recognized in today's cityscape (Figure 3).

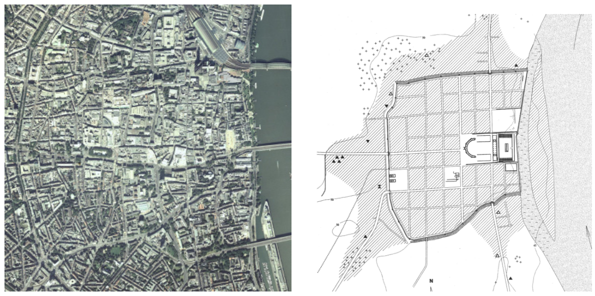

Figure 3: Aerial photographs of Cologne today (left) are used as one additional source for the reconstruction of the digital terrain model of the ancient town (right).

The reconstruction results represent the scientific fundament and are used by the design team to create the virtual 3D models. Therefore, archaeologists deliver all suitable data such as textual descriptions, photographs of artefacts, and highly detailed 2D computer aided design (CAD) drawings (in DWG file format) of building parts and their arrangement to the designers.

2.2 3D Model Creation

Based on this scientific evaluated material, designers create virtual 3D building models for the real-time visualization and Flash® content for the presentation of related information. Thereby, one challenge for designers is to balance the classical trade-off between visual quality and suitability for real-time rendering to meet the requirements of the two other groups. On one hand, archaeologists demand for maximum visual quality for every detail. On the other hand, computer graphic engineers rely on low polygonal representations for fast rendering on graphics hardware, to ensure interactivity for the expected number of high-detail building models.

The 2D CAD reconstruction drawings, imported into a 3D modeling tool, serve as blueprints to model single building pieces. Because buildings in Roman architecture can contain a large number of small ornamental elements, which would result in a huge number of polygons within the 3D reconstruction, designers have to work anticipatorily. To reduce the geometric complexity, rounded structures, represented by spline curves (Foley et al. 1990) in the CAD drawings, are approximated by polygonal counterparts. Generally, all individual objects are constructed with the least possible amount of polygons while remaining the original shape.

After a library of 3D elements (e.g., capitals, columns, doors) was created, these structures are combined to complete buildings and provided with 2D texture-maps representing the respective materials. These material textures are based on color templates prepared by the archaeologists. Further, static lighting conditions are assumed to enhance the 3D impression of the models and to accelerate rendering. Therefore, the material textures are combined with light maps, which are derived from global illumination simulations. Finally, all building positions are geo-referenced and scaled with respect to the overall 3D scene.

To exchange 3D models a data format has to be selected that meet the following three criteria:

1. It has to be supported by digital content creation (DCC) tools used by the designers (Cinema4D, 3DSMax).
2. To provide content preservation, the format should be extensible, future proof, and store information lossless.
3. The format should support a minimum set of standard features (e.g., geometric transformations, color and material definitions, geometry instancing, and external references for reusing building elements.

To enable the use of state-of-the-art rendering techniques supported by modern graphics hardware, e.g., multi-texturing or shader programs (Kessenich 2006), the support of related features by the format is additionally desirable.

We decided on using the *Collada* exchange format (Arnaud and Barnes 2006, Khronos Group 2008). It fits all of above requirements, is based on an XML scheme, has an open specification, and is supported by a large number of major 3D hardware and software companies.

2.3 Interactive 3D Visualization

Compared to static 2D illustrations, interactive 3D visualizations, presenting the Roman buildings within their context, enable archaeologists to discuss or validate their hypotheses with a direct access to the spatial situation. Thus, it become possible to analyze the arrangement of buildings related to the terrain, their mutual visibility from a pedestrian perspective, or the connectivity regarding to the path network. To support these tasks in virtual archeology applications, two conditions have to be considered:

1. Scientific users demand for free navigation, allowing them to inspect every detail in the complete 3D scene. They do not accept restrictions to particular areas or viewing angles, such as used for optimizations in 3D computer games.
2. The models delivered to the 3D CG engineers are created with standard DCC tools. Again, compared to specialized level editors or content creation tools, this allows only a limited set of optimizations during the modeling stage.

For these reasons, and because Collada is more a flexible exchange format than an efficient storage or rendering format, the 3D models have to be converted in a first step. Therefore, we developed a configurable converter tool chain that applies a set of optimizations to each model to prepare it for efficient real-time rendering. To minimize the loading times this converter uses a simple binary format for persistent storage of the optimized 3D model representations. Nevertheless, the Collada representations are not affected and stay preserved as input data for other visualization applications in the future.

Besides model optimization, the second task of the CG team is to develop an application framework for the interactive presentation, analysis, and exploration of the models. To support the need of different end-user groups, this team is responsible for permanent framework extension with enhanced visualization and adaptive navigation techniques.

3. OPTIMIZATION FOR REAL-TIME VISUALIZATION

3.1 3D Model Optimization

All 3D models pass a number of optimization steps to maximize their real-time rendering performance (Kuehne et al. 2005):

1. A *polygon cleanup* operator removes all obsolete data, such as unused texture coordinates, normal data, or vertex colors. Further, redundant information such as vertex duplicates or degenerated triangles are removed.
2. Next, elements of the scene graph are reordered to *minimize state switches* for the graphic hardware. Afterwards, polygonal representations that using the same material or texture form a group whose elements are rendered together in a sequence.
3. The third optimization adds an *index structure* to the polygonal representations if a large number of triangles sharing the same vertices coordinates. This results in a more compact representation and reduces the allocation of the limited graphic board memory.
   Afterwards, indices are reordered to *optimize the cache hits* during the vertex processing in the graphic processing unit.
4. In the fourth step, geometries for different objects are merged to batches with a maximum of 2^{16} vertices to improve the rendering throughput.
5. At least, hardware *texture compression* is applied for each texture to reduce the GPU memory consumption and accelerate their transfer from system memory to the graphics board.

3.2 Rendering Optimizations

To improve the rendering performance for a scene constructed out of a number of individual 3D models, a further set of optimizations is applied. Because, illumination was pre-calculated for static lighting and stored combined with material colours in the surface textures, lighting calculations are turned off during interactive rendering. Appropriate shader programs are used to minimize the processed set of vertex and fragment operations. These programs calculate only the model-view and projection transformations for each vertex, and use a single texture look-up to determine the visible colour and intensity for each drawn pixel.

To reduce the triangle count processed by the GPU per frame, we apply standard culling techniques (Akenine-Möller and Haines 2002) and introduce a simple but effective level-of-detail (LOD) mechanism. Thereby, highly-detailed parts of building geometry, e.g., capitals, are omitted during phases of intense user navigation. If the interaction with the scene stops, these elements are rendered subsequently to achieve a depiction containing all details.

4. VISUALIZATION FRAMEWORK

4.1 XML Configuration

Both, the tool chain for model optimization and the visualization application can be completely configured using XML files. This data-driven approach gives the flexibility to easily create and validate new optimization schemes or 3D visualizations scenarios without programming skills.

The XML file for the 3D visualization enables the definition of different visualization scenarios. For each scenario, multiple camera positions and orientations, the 3D models that should be displayed, and a set of navigation constraints can be defined. The camera settings describe predefined views to important elements of the scene. An animated camera movement can be derived by a consecutively interpolation between two camera settings. Navigation constraints help to avoid getting-lost situations. Therefore, a maximum distance to a point in the scene or a minimum height for the observer position, e.g., above the digital terrain, can be specified.

4.2 Communication with Flash Widgets

The 3D visualization can be controlled by other widgets which show additional multimedia information using Adobe® Flash®. This functionality was developed for the use in museum and edutainment applications. The communication between 3D visualization and widgets is based on a client-server architecture. Decoupling these widgets from the 3D visualization enables the independent creation and modification of additional content without side effects for our framework.

Figure 4: Screenshot of a flash widget that is used to control a 3D visualization screen for the presentation in a museum.

Figure 4 shows a prototype for a museum presentation, where Flash® content is presented on a separate computer with a touch screen interface. In addition to all presentation features provided by Flash®, the user can trigger commands from this screen to load a particular scene, or take the view of a predefined observer position in the 3D view. These commands are communicated using standardized network mechanisms. After such a command call, the 3D visualization informs the Flash® widget about the execution status of the triggered command.

5. ENHANCED VISUALIZATION TECHNIQUES

5.1 Cut-Away Views

Interactive cut-away views are an important visualization technique that reveals the interior of complex models by clipping either occluding parts or outer layers (Li et.al 2007). Usually, these depictions are static and where created by hand, thus the view point and the displayed cuts are fixed. Interactive cut-away views overcome these drawbacks by enabling the user to choose desired cut planes and volumes, while navigating in the virtual environment simultaneously.

To enable this functionality in our framework, we integrate a technique described in (Trapp and Döllner 2008). It supports clipping against a number of arbitrary solid volumes within a single rendering pass. The volume meshes can be designed using standard modelling tools and are positioned into the scene using the XML configuration file. At runtime, rasterized representations of these volumes are created in a pre-processing step. During rendering, every fragment is tested using a volumetric parity test with respect to these representations. The fragment is discarded if its associated position lies inside a specific volume.

Figure 5 (left) shows an example for applying two different cuts to a building model in real-time. It reveals parts of the building footprint and internal structures, such as walls and doorways, which otherwise would be hidden to the viewer. Figure 5 (right) shows a close up on a clipped column with applied capping to convey an impression of solid material on the clip surface.

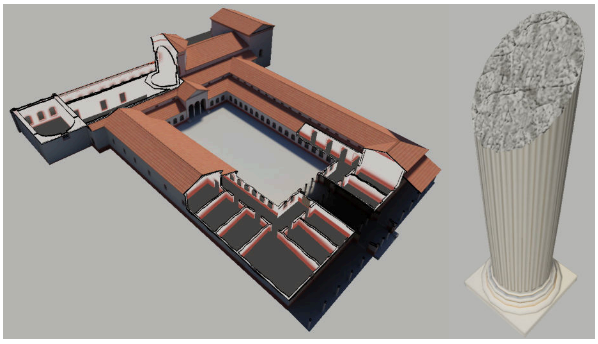

Figure 5: Left: Generated cut-away view of the Praetorium building model using two cut volumes. Right: Single column that is clipped and capped.

The quality of the 3D models, in terms of modelled interior, solid walls, and consistency of polygon orientation, is important for the resulting visual impression. Besides additional configuration issues, the usage of cut-away views introduce a number of challenges and technical implications to our visualization framework:

- **Shading Discontinuities:** Figure 5 (left) shows possible discontinuities in shading for areas inside and outside the building and shadows. This is caused by the pre-lighted object textures that are only valid for views from outside. This effect can be compensated partially by using image-based global lighting approaches that approximate ambient occlusion (Luft et.al 2006).

- **Performance Decrease:** The applied clipping technique is fill-limited, i.e. the runtime performance depends on the number of fragments tested against the volumetric raster representation. To avoid a significant performance decrease, this technique is currently applied to parts of the full scene.
- **Advanced Occlusion Culling:** To deal with the rendering of hidden but potential visible geometry, the framework has to apply more advanced occlusion culling mechanisms (Mattausch et al. 2008).

5.2 Automated Label Placement

Labels are used to communicate short textual information that can not be derived from the depiction of the 3D model. Typically, they identify object parts, show measurements, or give hints to secondary material, e.g., using website URLs. For single views and a small amount, labels are can be placed by hand. However, interactive environments call for an automated process that calculates adequate label positions in real-time. Thereby, each label should be placed readable and in a way that allow users to assign it with the related object unambiguously. Further, a label should neither overlap other labels nor important elements of the depiction and should not be occluded by elements closer to the observer.

We use the approach presented in (Maass and Döllner 2008) that seamlessly integrate labels into the 3D scene instead of placing them in a separate layer, where they superimpose the depiction (Figure 6). Therefore, hull bodies that generalize the complex object geometry are used to define all areas suitable for embedding a label. The simplest hull variant covers all large areas of a building with a set of rectangles. Sample points that serve as label candidate positions are equally distributed over these rectangles. To allow a fast determination which position at which rectangle provide the best quality for a given viewing direction, the visibility of hull bodies, hull elements, and sample points is tested hierarchically. For each rectangle a score for the orientation and visible area is calculated to sort them with respect to their quality potential. Afterwards, all rectangles are iterated using this order until a position that allows a visible label embedding is found.

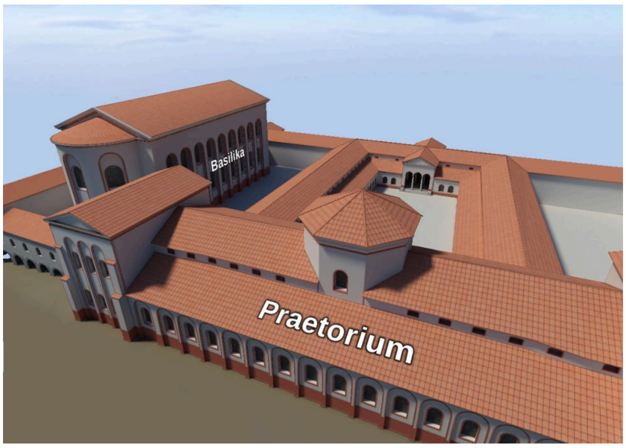

Figure 6: Embedded annotations dynamically placed at large areas that are oriented towards the viewer are used to communicate textual information about the building models.

5.3 Non-photorealistic Rendering

The style used to create an image has a huge effect on how the depiction is interpreted. Photorealistic renderings tempt people to believe they look at photographs of real world objects. For virtual archaeology this kind of presentation is often not sufficient. Archaeologists require the ability to express uncertainty such as missing evidence or accuracy in the reconstructions. Artist often use sketchiness in their illustrations to communicate such uncertainty. In computer graphics the generation of stylistic drawings belongs to the area of non-photorealistic rendering (NPR). A large number of different NPR techniques have been published in the research literature. An overview can be found in the textbooks by (Gooch and Gooch 2001) and (Strothotte and Schlechtweg 2002).

In (Roussou et al. 2003) watercolor and pen-and-ink representations are used to communicate uncertainty in archaeological models. For our purpose, it is important that the NPR technique can be applied in real-time and that the integration into our framework can be done without major structural changes. Therefore, we choose an image based abstraction technique that can be applied as a post-processing effect to the intermediate photorealistic rendering result (Kyprianidis and Döllner 2008). The technique smoothes low-contrast regions while preserving edges and enhance salient important edges (Figure 7).

Figure 7: 3D model of a temple. Top: Photorealistic rendering. Bottom: NPR rendering to communicate missing evidence.

6. RESULTS AND OUTLOOK

The established content creation process as well as the chosen data formats proven themselves in practice. The developed framework allows the interactive exploration of the virtual reconstructed Roman Cologne. Figure 10 shows a screenshot from the interactive 3D visualisation containing all currently reconstructed buildings and the terrain model. For most of them, modelling, texturing, and lighting simulation is finished. A complete list with the geometric complexity and the amount of RGB texels used for each model is given in Table 8. Figures 11-13 show additional screenshots from the real-time visualization to give an impression about the high visual quality of the 3D building models.

Model	Complexity		
	#Vertices	#Triangles	#Texels
Ara Ubiorum	1,652,996	2,197,182	49,670,283
Cryptoporticus	1,477,628	1,890,087	49,670,283
Dionysos Villa	143,296	236,276	33,113,522
Forum Basilica	438,132	521,718	16,556,761
Insulae	350,298	522,662	62,694,105
Praetorium	416,822	600,494	74,615,652
Round Temple	379,655	540,655	49,670,283
Temple	2,378,845	3,210,162	66,227,044
Market Halls	237,648	269,912	16,556,761
Terrain	52,811	92,223	33,113,522
City Wall	248,672	483,174	0
Sum	7,776,803	10,564,545	451,888,216

Table 8: Geometric and visual complexity of the 3D models used in the interactive visualization.

The applied optimizations enable our framework to render the complete model (without omitting high-detailed building parts) at interactive frame rates. Table 9 shows our measurements on four different test systems.

Configuration	Screen Resolution	FPS
Athlon 64 X2 Dual-Core 4200+ 2.21 GHz, 2GB RAM, GeForce 8800 GTS 640 MB	1600x1200	20
Intel Xeon E5345 2.33 GHz, 3GB RAM, GeForce 8800 GTX 512 MB	1600x1200	20
Intel Core 2 Duo X6800, 2.93 GHz, 3 GB RAM, GeForce 7950 GT 512 MB	1600x1200	12
Intel Core 2 Duo T7700 2.4 GHz, 3GB RAM, NVidia Quadro FX570M 128 MB	1400x1050	8

Table 9: Average rendering frame rate per second (FPS) measured on three different test systems for the whole scene of Roman Cologne.

As next step, we are going to complete the city of Roman Cologne. For the future, it is planned to further refine and extend the city model. One concrete idea is the reconstruction of the street of tombs, which was located outside the gates of the city walls. However, with increasing model complexity, we have to switch to out-of-core rendering techniques (Dietrich et al. 2007), because it is currently near to the limits of today's graphics hardware. Furthermore, we plan extend our real-time visualization with high dynamic range light simulation (Kuchar et al. 2007) to allow an adapted presentation of interior building structures.

Additionally, we want to make the huge amount of existing scientific material accessible within the project. Therefore, we plan to extent the framework to interlink elements of the buildings with material from existing scientific data bases, e.g. the *Arachne* database (www.arachne.uni-koeln.de).

ACKNOWLEDGEMENTS

This work has been funded by the German Federal Ministry of Education and Research (BMBF) as part of the InnoProfile research group '3D Geoinformation' (www.3dgi.de).

REFERENCES

Almagro, A., Rodríguez, C., González, M., Zúñiga, I., 2006. The Alcazar of Seville in the 14th Century, *7th Symposium on Virtual Reality, Archaeology and Cultural Heritatge (VAST)*, pp. 15-22.

Akenine-Möller, T., Haines, E., 2002. *Real-Time Rendering*, 2. ed.. A K Peters Ltd., Wellesley, Mass..

Arnaud, R., Barnes, M. C., 2006. *Collada - Sailing the Gulf of 3D Digital Content Creation*. A.K. Peters Ltd., Wesley Mass..

Debevec, P. E., 2005. Making "the parthenon", *Invited paper of VAST 2005*, Pisa, Italy. http://www.debevec.org/Parthenon/debevec-vast2005-parthenon.pdf.

Dietrich, A., Gobetti, E., Yoon, S.-E., 2007. Massive-Model Rendering Techniques., *IEEE Computer Graphics and Applications*, 27(6), November/December 2007, pp. 20-34.

Foley, J. D., van Dam, A., Feiner, S. K., Hughes, J. F., 1990. *Computer Graphics: Principles and Practice*, 2nd ed., Addison-Wesley Professional.

Gooch, B., Gooch, A. A., 2001. *Non-Photorealistic Rendering*, AK-Peters.

Irmler, B., 2004. Rekonstruktion einer Porticusordnung an der Rheinseite in Köln. 77–102, *Kölner Jahrbuch 37*, Gebr. Mann Verlag, Berlin.

Kessenich, J., 2006. *The OpenGL Shading Language*, Version: 1.20 Document Revision: 8, Manual, September 2006.

Khronos Group, 2008. *Collada 3D Asset Exchange Format* http://www.khronos.org/collada/, (accessed 20 Aug. 2008).

Kuchar R., Schairer T. and Straßer W., 2007. Photorealistic Real-Time Visualization of Cultural Heritage: A Case Study of Friedrichsburg Castle in Germany, in: *Eurographics Cultural Heritage Papers*, pp. 9-16.

Kuehne, B., True, T., Commike, A., and Shreiner, D., 2005. Performance OpenGL: Platform Independent Techniques. In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California, July

31 - August 04, 2005). J. Fujii, Ed. SIGGRAPH '05. ACM, New York, NY, 1.

Kyprianidis, J. E., Döllner, J., 2008. Image Abstraction by Structure Adaptive Filtering, *Proc. EG UK Theory and Practice of Computer Graphics*, pp. 51–58.

Li, W., Ritter, L., Agrawala, M., Curless, B., Salesin, D, 2007. Interactive Cutaway Illustrations of Complex 3D Models, *Proc.s of the ACM SIGGRAPH Conference* 2007, pp 31.

Luft, T., Colditz, C., Deussen, O., 2006. Image enhancement by unsharp masking the depth buffer, *Proc. of the ACM SIGGRAPH Conference* 2006, pp. 1206-1213.

Santos, L. P., Coelho, V., Bernardes, P., Proença, A., 2006. High Fidelity Walkthroughs in Archaeology Sites, *6th Int. Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*, Pisa, Italy, pp. 453-458.

Maass, S., Döllner, J., 2008. Seamless Integration of Labels into Interactive Virtual 3D Environments Using Parameterized Hulls, *4th International Symposium on Computational Aesthetics in Graphics*, Lisbon, pp. 33-40.

Mattausch, O., Bittner, J., Wimmer, M., 2008. *C*HC++: Coherent Hierarchical Culling Revisited, *Computer Graphics Forum*, 27(2), pp. 221-230.

Rost, R. J., 2006. *OpenGL Shading Language*, 2. ed.. Addison-Wesley, London.

Roussou, M., George Drettakis, G., 2003. Photorealism and Non-Photorealism in Virtual Heritage Representation. , *Proc. of the 4th Int. Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*, Eurographics, pp. 51-60.

Strotthotte, T., Schlechtweg, S., 2002. *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*, Morgan Kaufmann.

Trapp, M., Döllner J., 2008. Real-Time Volumetric Tests Using Layered Depth Images*, Proc. of Eurographics* , pp. 235-238.

Weibel, R. Heller, M., 1991. Digital terrain modelling, In D.J. Maguire, editor, *Geographical Information Systems: Principles and Applications*, Longman Scientific & Technical, pp. 17–90.

Figure 10: Overview of the 3D visualization containing all currently reconstructed buildings of Roman Cologne.

Figure 11: Screenshot from the real-time visualization with a view from within the Praetorium model.

Figure 12: Screenshot from the real-time visualization showing the details of the Ara Ubiorum model.

Figure 13: Screenshot from the real-time visualization showing the details of the Cryptoporticus model.